Report Number: WUCSE-2004-49

2004-08-10

Road Extraction From Aerial Video Using Active Contours and Motion Cues

David A. Jurgens

This thesis motivates a fully automatic approach for locating roads in aerial video by using active contours in conjunction with motion cues. Directed active contours provide ideal parametric representations of roads due to their ability to fit the variety seen in road shapes. Motion in stabilized aerial video can be represented as a distribution of spatio-temporal derivatives. These motion cues can then be incorporated into the energy function, which yields active contours that better model the active roads in a scene. We present results using this approach on typical models of both urban and rural scenes.

... Read complete abstract on page 2.

Road Extraction From Aerial Video Using Active Contours and Motion Cues

David A. Jurgens

Complete Abstract:

This thesis motivates a fully automatic approach for locating roads in aerial video by using active contours in conjunction with motion cues. Directed active contours provide ideal parametric representations of roads due to their ability to fit the variety seen in road shapes. Motion in stabilized aerial video can be represented as a distribution of spatio-temporal derivatives. These motion cues can then be incorporated into the energy function, which yields active contours that better model the active roads in a scene. We present results using this approach on typical models of both urban and rural scenes.

Department of Computer Science & Engineering

Washington University in St. Louis
School of Engineering & Applied Science

2004-49

Road Extraction From Aerial Video Using Active Contours and Motion Cues, Master's Thesis, August 2004

Authors: Jurgens, David A.

August 10, 2004

Abstract: This thesis motivates a fully automatic approach for locating roads in aerial video by using active contours in conjunction with motion cues. Directed active contours provide ideal parametric representations of roads due to their ability to fit the variety seen in road shapes. Motion in stabilized aerial video can be represented as a distribution of spatio-temporal derivatives. These motion cues can then be incorporated into the energy function, which yields active contours that better model the active roads in a scene. We present results using this approach on typical models of both urban and rural scenes.

SEVER INSTITUTE OF TECHNOLOGY

MASTER OF SCIENCE DEGREE

THESIS ACCEPTANCE

DATE: August 10th, 2004

STUDENT'S NAME: David A. Jurgens

This student's thesis, entitled Road Extraction From Aerial Video Using Active Contours and Motion Cues has been examined by the undersigned committee of five faculty members and has received full approval for acceptance in partial fulfillment of the requirements for the degree Master of Science.

APPROVAL: _____ Chairman

_____

_____

_____

_____

WASHINGTON UNIVERSITY
SEVER INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

ROAD EXTRACTION FROM AERIAL VIDEO USING ACTIVE CONTOURS AND MOTION CUES

by

David A. Jurgens

Prepared under the direction of Professor Robert Pless

A thesis presented to the Sever Institute of
Washington University in partial fulfillment
of the requirements for the degree of

Master of Science

December, 2004

Saint Louis, Missouri

WASHINGTON UNIVERSITY
SEVER INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

ABSTRACT

ROAD EXTRACTION FROM AERIAL VIDEO USING ACTIVE CONTOURS AND MOTION CUES

by David A. Jurgens

ADVISOR: Professor Robert Pless

December, 2004
Saint Louis, Missouri

This thesis motivates a fully automatic approach for locating roads in aerial video by using active contours in conjunction with motion cues. Directed active contours provide ideal parametric representations of roads due to their ability to fit the variety seen in road shapes.. Motion in stabilized aerial video can be represented as a distribution of spatio-temporal derivatives. These motion cues can then be incorporated into the energy function, which yields active contours that better model the active roads in a scene. We present results using this approach on typical models of both urban and rural scenes.

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank professor Robert Pless for his patience, enthusiasm and insight into how research should be done. I would also like to thank the Waldemar Estate for its continued dedication to excellence, and thank my parents for their support.

David A. Jurgens

Washington University in Saint Louis
December 2004

Chapter 1

Introduction

Comprehensive road databases are essential for transportation planning and efficiency. In places where no database exists, or in areas where sufficient construction has taken place to alter the road outlines, a manual effort must be made to construct accurate representations of the road network, which for large databases can be tedious and often difficult. For this reason, an effort has been made to partially or fully automate the process of creating a road database from aerial video or image data. However, the highly textured and varied nature of aerial imagery makes correct road extraction difficult to automate. The dissimilar nature of urban and rural scenery makes using contextual information problematic without targeting a specific landscape or having a large database of possible objects in aerial imagery. Despite such difficulties, the potential benefits of automatic road extraction are enough to prompt research for over twenty years.

1.1 Motivation

The Computer Vision and Geographic Information Systems communities have put forth a multitude of techniques for automatic or semi-automatic road extraction. Due to the availability of aerial imagery, such as the IKONOS satellite images, most algorithms utilize information from images. Only recently have computational and storage capabilities made the analysis of aerial video possible.

The use of single images ignores the fact that motion is one of the fundamental features of road scenes. A defining feature of active road scenes is motion from moving

vehicles, and capturing a representation of the motion requires video. Increases in the resolution of aerial cameras and video recording devices now allows video to show the fine detail necessary for frame to frame motion detection of individual vehicles. Furthermore, the additional space constraint for using video has been eliminated by capturing a statistical representation of motion in the scene without the need for the video to persist in memory. Both Dai et al. [2] and Pless et al. [16] describe techniques to capture the distribution of spatio-temporal derivatives for each pixel by retaining only filter responses, thus allowing the video data to be discarded. The distributions are a compact representation that can be used to summarize local motion at a pixel.

1.2 Overview

The approach in this thesis uses the representation of motion captured by distributions of spatio-temporal derivatives as the basis for extracting roads from aerial video. We first present an overview of relevant work in the field of road extraction as well as spatio-temporal derivatives. Following, we describe the methods of extracting spatio-temporal distributions from aerial video, and then discuss the post-processing steps to remove erroneous samples from the collection of distributions. Next we describe the technique for using active contours to represent road features. Lastly, we show and discuss results from road extraction using this approach.

Chapter 2

Relevant Prior Work

2.1 Road Extraction

In contrast to our analysis of video sequences, nearly all prior work in visual road extraction considers the analysis of static aerial or satellite imagery. These approaches can be characterized by the local image processing they perform, the manner in which they define or build road descriptions, and the form of the final model.

Approaches that create pixel based maps defining road segments on the image must distinguish between road pixels and their surroundings. Often in aerial imagery, roads and surroundings have similar appearance and color. Geman and Jedanyk cite this as the reason for using local functions that search for arcs where pixel intensities are more similar along the arc and where surrounding areas are different from the arc [4]. They then use these function responses to grow roads as a probabilistic tree which is later pruned to give a formal road specification in terms of arc segments. Hinz adds to this approach by using a mixed-scope method; analyzed region features are broken into different contexts such as urban, rural and forest and then used as guides to how to do local analysis at the pixel level [5]. The local analysis is done by constructing road segments from small features and then connecting them based on a probabilistic model. This connection strategy succeeds in bridging occluded or obscured road segments where the appearance of the road is not consistent. Bicego expands upon this approach by including color and gradient information to extend

road segments [1], and Porikli uses local filters to examine road probability and direction, thereby allowing uncategorized pixels to be included provided they have the right direction [17].

Park uses a different model from that of Hinz by utilizing templates to identify road segments and then continually shifting and rotating a recognized template to expand segments [15]. Price also uses templates but, similar to the mixed-scale method of Hinz, templates are used to identify intersections, and then roads are expanded using local models [18]. Kumagi takes the opposite approach by using only global features. Roads are extracted from IKONOS images by examining spectral reflectance data to eliminate vegetative reflectances and then extracting roads from histogrammed models of the remaining reflectances using a maximum *a postireori* probability hypothesis [11].

Instead of finding pixels of local segments that lie along roads, it is often useful to have a description of overall road shape. Snakes provide one such representation. Kass et al. define active contours as a method for describing a parametric curve that fits to features that minimize the energy of the contour as defined by its energy function [9]. In implementation, a contour is comprised of a set of control points with cubic splines connecting control points to form the curve. Any approach that uses active contours for road extraction must then define the two critical issues of placing (i.e. seeding) the initial control points and defining an energy function that is minimal when the snake lies along the road.

Huber and Lange use a genetic algorithm to select points on the active contour [7]. Though not using snakes, Lin and Chen promote a method for finding control points by using Sobel filters and then doing post processing to find intersections, ultimately forming roads by combining recognized "meaningful regions" [14].

Huber defines the following energy function,

$$E = \sum_{i=1}^{n} (\alpha E_{cnt}(\upsilon_i) + \beta E_{crv}(\upsilon_i) + \gamma E_{img}(\upsilon_i) + \zeta E_{cnv}(\upsilon_i)) \tag{2.1}$$

where υ_i denotes the i^{th} control point of the snake. This formulation uses the standard definitions of curvature energy for E_{crv} and E_{cnv}, which tends to move points toward a straight line. The standard formulation for expansion for E_{cnt} requires that all points maintain a minimum distance from each other. E_{img} is calculated by computing the

image gradients with respect to the x-axis; and y-axis. α, β, γ and ζ denote weighting terms and define the importance of each of the forces in adjusting the position of the snake. Laptev et al. describe a similar energy function written more explicitly for the ribbon snakes, which represent both the trajectory and width of a curve:

$$E(\vec{v}) = -\int_0^1 P(\vec{v}(s))ds + \frac{1}{2}\int_0^1 \alpha(s)|\frac{\partial\vec{v}(s)}{\partial s}|^2 + \beta(s)|\frac{\partial^2\vec{v}(s)}{\partial s^2}|^2 ds. \tag{2.2}$$

The image energy, $P(\vec{v}(s))ds$, is defined to be the energy from each side of the ribbon snake which is defined as $(\nabla I(\vec{v_L}(s)) - \nabla I(\vec{v_R}(s))) \cdot \vec{n}(s)$, and where s is the arc length parameter defining distance along the snake [12]. This approach uses the same formulation for bending energy as that of Huber. However, while still using the image gradient as the source for image energy, this approach defines energy with respect to a snake with a width. Laptev et al. contrast from Huber in the manner of updating control points. Laptev et al. use a technique called "ziplock snakes" in which control points in the middle of the contour are only affected by shape forces (as opposed to image forces), which in Equation 2.1 would mean that $\gamma = 0$, and in Equation 2.2 the energy from $P(\vec{v}(s)ds)$ is discounted. The intent of such a formulation is that once outer control points locate significant image features, the center control points fit to a predefined shape. In the context of road extraction, outer control points locate the ends of roads and central points continue the curve regardless of image energy, which allows for a smoother curve if occluded regions exist in the middle segments of the road. Furthermore, the preprocessing described in Laptev [12] is analogous to that of Hinz [5] in that both use coarse scale features as input for analysis of fine scale features.

2.2 Problems with Prior Methods

Hinz points out that many of the approaches for road extraction are focused on either rural imagery or urban imagery [5]. This specialization is in part due to the different nature of the two scenes. The inability of most road extraction approaches to fully extract roads in both scenes exemplifies the fragility of these approaches. Scene-specific customization is often necessary for quality results from static image extraction.

The use of "ziplock snakes" as described by Laptev may also fail to fully extract roads with high local curvature. As outer control points converge to road features, central points may not correctly converge to smaller curved segments if no image energy is applied to them. While this could be corrected by an analysis of how much of the snake should be passive, the approach still would require extra heuristics to ensure better road extraction. These heuristics imply that the snake model proposed by Laptev is not able to formulate a general model for all roads in aerial imagery.

2.3 Spatio-Temporal Derivatives

The use of spatio-temporal derivatives in active contours has not been explored until recently due to space and processing requirements. Using the same filtering technique described in this thesis, Kühne et al. use a spatio-temporal tensor field as the main component of the energy function for object segmentation with active contours in video [10]. This approach uses data similar to the rooftop mall scene described in 5.1.3 to build spatio-temporal derivative distributions. These distributions are then used with active contours to extraction moving automobiles from active traffic scenes. To our knowledge no one has used spatio-temporal derivatives for road extraction.

Chapter 3

Spatio-Temporal Derivatives as Motion Cues

This approach uses spatio-temporal derivatives to define roads as contours in geo-rectified aerial video. This differs from previous works which use feature recognition to assemble roads from aerial imagery; the approach described in this thesis specifically does no such feature tracking or recognition. Here we take an approach that considers video as a 3D function describing the intensity of a pixel at (x, y, t) where x and y denote a pixel location on a frame and t denotes the time. Although the intensity function, $I(x, y, t)$, is only measured at discrete values of x, y, t, it is often useful to consider this as a continuous function. The derivatives of $I(x, y, t)$ form the basis for modeling motion in the approach described in this thesis.

3.1 Calculating Spatio-Temporal Derivatives

For each pixel in the image, intensity derivatives are calculated yielding the function of $I(x, y, t)$. These derivatives are calculated using specialized blurring and derivative filters, as described by Farid, which yield better results than applying Sobel filters, such as the one shown in Figure 3.1, to raw images [3]. For each pixel (x, y, t), a filter with respect to the x-axis, y-axis and time is applied. The derivatives calculated by the filter are denoted as I_x, I_y and I_t.

-1	0	1
-2	0	2
-1	0	1

Figure 3.1: The Sobel filter for calculating derivatives along the x-axis

3.2 Optic Flow

The spatio-temporal derivatives of $I(x, y, t)$ provide an excellent way to describe motion at each pixel. Formally, for a pixel with location (x, y) at time t, we may define $I_x(x, y, t)$ as the derivative of the image intensity with respect to the x-axis of the image, and $I_y(x, y, t)$ as the derivative with respect to the y-axis. $I_t(x, y, t)$ is defined as the change in intensity through time. By collecting these through time, we are able to build a distribution model for each of the spatio-temporal derivatives. We represent this distribution as the covariance matrix Σ. Storing the distribution allows us to discard the derivative values and video frames, thereby removing the constraint of keeping large portions of the video in memory. We discuss in detail features of Σ after an explanation of how to calculate the spatio-temporal derivatives and optic flow.

Horn and Schunck describe a classic equation for 2D motion at a pixel (x, y, t); the relationship between the optic flow and I_x, I_y and I_t is [6] :

$$I_x u + I_y v + I_t = 0 \tag{3.1}$$

where u and v denote the average speeds and directions of objects moving across the pixel at x,y. This equation holds true in video sequences where the change in motion is small and varies smoothly. It is not possible to directly solve this equation for the optic flow since it has two unknowns (u, v). To accommodate this problem, many algorithms add the constraint that the optic flow remains constant across local areas and then collect I_x, I_y and I_t derivatives to solve for the best optic flow. However this assumption does not hold true around the edges of moving objects, which ultimately results in incorrect motion estimates. In this approach, we consider a static camera allowing constraints on the optic flow to be combined through time. Placing the constraint through time allows for a more accurate optic flow to be computed when there are consistent motions in the scene.

Video allows for optic flow to be constrained in the time dimension. Aerial video, where a camera on an airplane records video of the ground plane, provides a convenient manner of collecting relevant data. In these videos, the plane often moves relative to the ground, causing a global transform of the video images. These transforms invalidate the assumption that an image pixel views the same geographic location in a scene. Therefore video must be geo-referenced (or ortho-referenced) such that geographic locations maintain the same position in the video throughout the segment.

Geo-referenced aerial video allows for the combination of optic flow equations at each pixel through time. Because the constraints are only in the time dimension, the optical flow can be computed without the need for information from an extended spatial region. Because each pixel shows the intensity of an unmoving geographic location, we expect a consistent motion pattern through that pixel, where motion is seen as changes in intensity. These derivatives describe the type of motion at that pixel. Because we assume that all motion is of a constant type, we can assert that each frame represents a sample from the distribution that represents the true motion at that pixel. We store the distribution of $< I_x, I_y, I_t >$ as the covariance matrix Σ.

This distribution is modeled as Gaussian distribution - although such a model is not necessarily accurate - which is stored as the covariance matrix $Sigma$. As new frames are processed, each of the free parameters of Σ are updated. Because Σ is symmetric, only six parameters are necessary. We formally define Σ as $\begin{bmatrix} I_x^2 & I_xI_y & I_xI_t \\ I_xI_y & I_y^2 & I_yI_t \\ I_xI_t & I_yI_t & I_t^2 \end{bmatrix}$.

3.3 Properties of the Covariance Matrix

The eigenvectors of the covariance matrix exhibit several interesting properties in relation to the motion at that pixel. If the eigenvectors of Σ, (v_1, v_2, v_3) with corresponding eigenvalues of (e_1, e_2, e_3), are sorted with the largest magnitude first, the following properties hold:

1. The vector v_3 is a homogeneous representation of the total least squares solution for the optic flow [8]. This is to say that the total least squares solution for optic flow is $\frac{v_3(1)}{v_3(3)}, \frac{v_3(2)}{v_3(3)}$. We call this 2D vector (f_x, f_y) for flow.

2. If, for all the data at that pixel, the set of image intensity derivatives exactly fits some particular optic flow, then e_3 is zero.

3. If, for all the data at that pixel, the image gradient is always in exactly the same direction, then e_2 is zero, which is the manifestation of the aperture problem. This implies that multiple optic flow solutions are equally consistent with the image derivatives.

4. The value $(1 - \frac{e_3}{e_2})$ varies from 0 to 1, and is an indicator of how consistent the image gradients are with the best fitting optic flow, with 1 indicating a perfect fit and 0 indicating that many measurements do not fit this optic flow.

5. The ratio $\frac{e_2}{e_1}$ varies from 0 to 1, and is an indicator of the specificity of the optic flow vector (f_x, f_y). If the ratio is near 0, then a number of optic flow vectors could fit the data of Σ; if it is close to 1, then the best fitting optic flow is more constrained.

Using the least squares solution, v_3, a vector field (u, v) of the optic flow is produced. The 2D vector field defined from (f_x, f_y) is a global description of the most probable optic flow at each pixel. These vectors as well as the consistency and specificity measures provide the basis for motion analysis in this approach.

3.4 Post Processing Techniques

Several heuristics also improve the identification of roads in aerial video. These attempt to mitigate motions due to poor image stabilization, or residual motion from objects not on the ground plane, which are not in the same plane and thus cannot be geo-rectified simultaneously.

3.4.1 I_t Filtering

Throughout video runtime, data is collected for aggregate I_t values. During the collection process the maximum I_t value for a pixel (x, y) is recorded if I_t is greater than a preset threshold. After empirical testing over several test sets, a value range of

Table 3.1: I_t threshold levels for data sets described in Section 5.1

Urban Video	20
Rural Video	15
Rooftop Mall Video	15

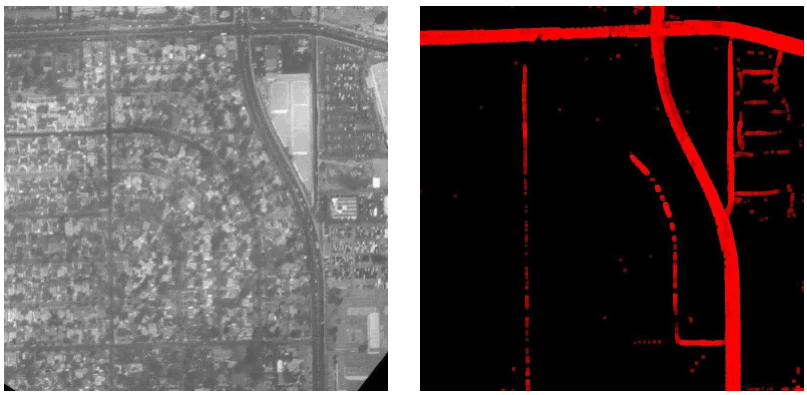
$15 - 20$ appears to produce quality results. The following table shows the thresholds for the test sets used in this paper:

Automating this process to set the threshold levels, such as basing the threshold on the distribution of I_t, remains a subject for further exploration. The threshold provides a method of separating regions of object motion from errors in intensity changes caused by problems such as those mentioned above. Although some good data is lost by thresholding, the process greatly reduces the amount of noise present in later data analysis. Collecting this data is necessary to use as a mask or filter for pixels at which the covariance data accurately represents the local motion at places in the image.

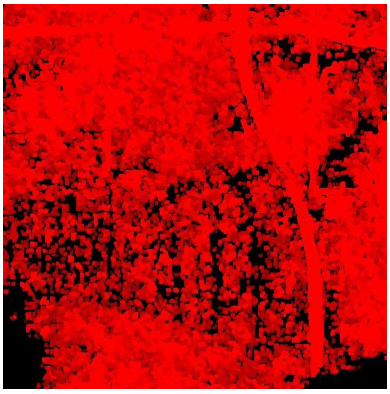
3.4.2 Motion-Oriented Covariance Matrix Smoothing

The optic flow solution from the spatio-temporal distribution is inconsistent due to the uneven distribution of data. In local regions, distributions model different motion because of errors in building the distribution. In imagery, blurring reduces errors by averaging out the effects of noise. However, radial Gaussian blurring of the tensor field produces more errors because it averages together data from different motion directions. For this reason, we use directional blurring; tensor fields are smoothed in the direction of the consistent motion.

Motion oriented smoothing uses the I_t filter point set to smooth only pixels with have significant motion. This reduces errors from tensor fields with inconsistent motion being smoothed with consistent matrices. The optic flow for each matrix $\Sigma(x, y)$ is used to form a directed region around the pixel at (x, y). Inside this region, the values of $\Sigma(x, y)$ are summed with those of $\Sigma(i, j)$ where i, j are in the region of influence. Influence is weighted based on distance and orientation of the optic flow at x, y. By recomputing the field in this manner, small gaps or anomalies in regions of consistent motion are smoothed out by averaging that data that is most likely to be similar. Formally, the optic flow field is recomputed using the following method:



a b



c

Figure 3.2: Images depicting the I_t filtering. Black edges denote errors from geo-rectification. (a) normal frame (b) using an I_t threshold of 20 (c) using an I_t threshold of 5.

1. For the covariance matrix $\Sigma(x, y)$ that has a recorded I_t above the threshold, let the optic flow at (x, y) be denoted as $< f_x(x, y), f_y(x, y) >$, where f_x and f_y denote the values of the optic flow for each axis.

2. Let $< d_x, d_y >= \frac{<f_x(x,y), f_y(x,y)>}{\sqrt{f_x^2(x,y)+f_y^2(x,y)}}$.

3. Let $T = \begin{bmatrix} 30d_x & 30d_y \\ -3d_y & 3d_x \end{bmatrix}$.

4. Let $M = T^\top T$.

5. Using these constructs, an output matrix $\hat{\Sigma}(x, y)$ is produced where $\hat{\Sigma}(x, y)$ is the weighted sum of neighboring covariance matrices. By defining T as such, the matrices oriented in the direction of (f_x, f_y) will have a higher weight.

6. Formally, for $\Sigma(i, j)$, a covariance matrix located at position (i, j) relative to (x, y), the weight is defined as

$$w(i, j) = e^{-<i,j>^\top M^{-1} <i,j>}.$$

7. The new smoothed covariance matrix is computed by

$$\hat{\Sigma}(x, y) = \sum_{i=-5}^{5} \sum_{j=-5}^{5} w(i, j)\Sigma(x + i, y + j).$$

Figure 3.3 illustrates the effects of motion oriented Gaussian smoothing. The bottom row of images shows the significant reduction in noise in areas of consistent motion. In areas where few cars have passed the smoothing also has a noticeable effect of unifying direction. Note that because the smoothing is motion oriented, no blurring occurs where two opposing motion fields are very close.

By combining both the I_t mask and the smoothed optic flow computed from $\hat{\Sigma}$, we are able to improve estimations of the motion of a scene.

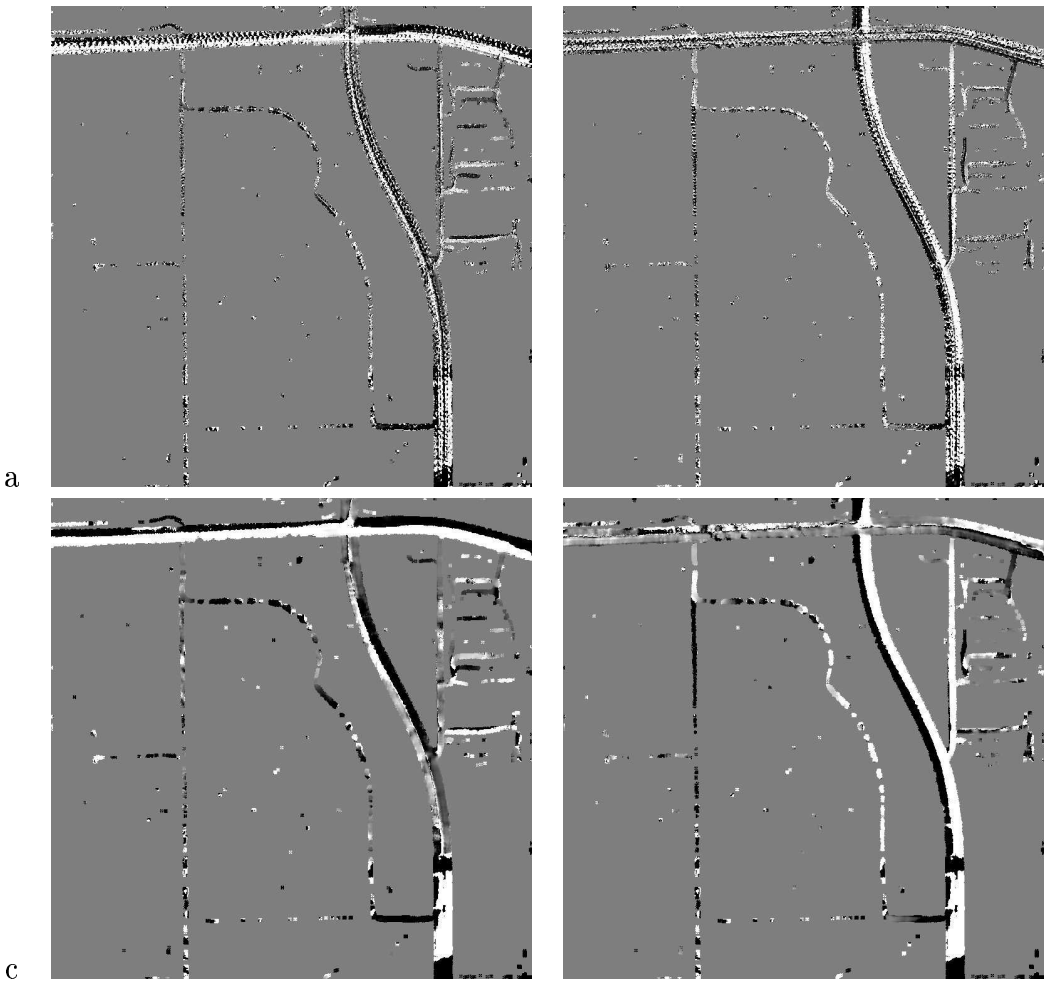


Figure 3.3: Urban video sequence images pre- and post-Gaussian smoothing. Images depict the optic flow at points where the I_t mask is non-zero; all other points are depicted using a mean-grey value. (a)the x-component of the optic flow pre-blurring (b) the y-component of the optic flow pre-blurring (c) the x-component of the optic flow post-blurring (d) the y-component of the optic flow post-blurring

Chapter 4

Defining Road Contours from Snakes

Given that roads come in a variety of shapes and orientations but are typically smooth curves, active contours provide an excellent representation for roads. While some approaches are only able to recognize road features for a preset number of templates, a snake contour can be fit to any type of curve and therefore are an easily adaptable model for the highly varying road shapes. Formally, a snake can be defined as set of control points through which the curve passes. A snake definition requires choosing the initial location of the control points and defining the energy function for updating their location. We first present a method for initializing snakes based on the optic flow vector field. Then we discuss defining snake energy based on snake direction and the optic flow. Lastly, we discuss minimization and snake fitness.

4.1 Initial Control Point Placement

The initial point selection is the initialization step and is critical to the success of road extraction. Although section 4.2 discusses the importance of the energy function, a good initialization step will choose points more likely to be on roads and thereby reduce the time to convergence. To improve road identification, control points are therefore drawn only from the point set of the I_t mask, which is assumed for the rest of the discussion in this section. We propose a progressive initialization that starts

with a single point and attempts to add additional points using the direction of the optic flow at the previous point.

First a seed point is randomly chosen. Then, the next control point is selected as the point at the head of the longest vector in the same direction as the optic flow that is still in the point set. If no point exists along the scaled optic flow vector that satisfies the condition, then an arc with the previous control point as the center is constructed at pixel intervals along the optic flow, and the furthest point in the point set with the least angular distance from the optic flow vector is selected as the next control point. The maximum deviance from the optic flow is $\frac{\pi}{4}$ radians. An additional constraint is added to selection; the dot product of the optic flow vectors at the control point and the point that is being tested can never be less than .75. This constraint ensures that the snake will not grow backwards, as it could when initialized in an area with two opposite, adjacent optic flows. Lastly, note that if during the seeding process, no point is found that has the aforementioned criteria, then the next control point will be set to the same location of the previous control point. This has the effect of stunting the growth of snakes that are do not have valid data, and provides a useful fitness heuristic that will be discussed later.

Selecting control points on the basis of optic flow provides several benefits. Variable extension allows the control points to to be placed across gaps in the I_t mask. Furthermore, placement based on direction allows the snake to follow curves and bends in the road without the need for image-based heuristics.

A snake is constructed from a fixed number of control points. If snake construction yields a snake with only a few control points with unique locations, then the snake as a whole is considered invalid. After empirical testing, we have found that a threshold of $\frac{1}{2}$ effectively eliminates snakes which would not fit to roads once optimized. If a snake is considered invalid, which becomes increasingly common as potential control points are removed from the set, then points in the region around the invalid snake are removed from the set. This prevents new snakes from forming in that region, which would be a wasted computation. Furthermore, we impose an additional heuristic where if the the number of co-located points is greater than or equal to $\frac{3}{4}$ of the snakes total number of points, then a larger area surrounding the snake is removed from the set of potential seed points. No new snakes are added after no points remain to seed another snake.

4.2 Specification of Energy Function

The energy function defines a trade-off between the shape of the snake and the fit to the features on the image. For a snake with a continuous length from 0 to 1, we define the energy function as:

$$E(\vec{v}) = \int_0^1 \alpha E_{img}(\vec{v}(s)) + \beta|\frac{\partial \vec{v}(s)}{\partial s}|^2 + \gamma|\frac{\partial^2 \vec{v}(s)}{\partial s^2}|^2.$$

where the first term denotes the energy from image features and the second and third terms denote the bending and expansion energies, which are contour features. To match the natural shape of the road, we use the bending energy to apply pressure toward straighter contours. Furthermore, we also use a high expansion energy to push control points out of local minima. In the implementation, we set $\alpha = .75$, $\beta = .5$, and $\gamma = 1$.

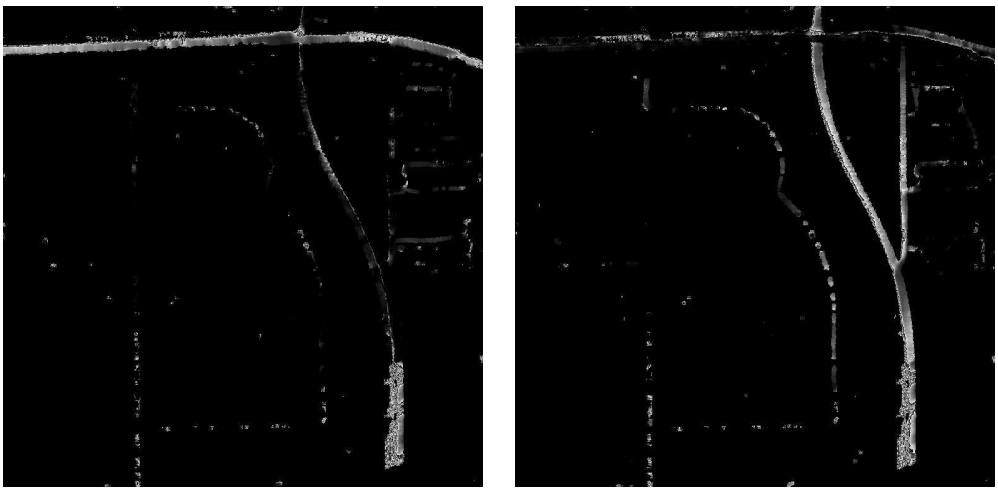
Ideally a snake fits to each road in the image. However, due a large search space, minimization of large snakes to accurate road models is computationally infeasible. Therefore, we present an optimization procedure and snake fitness function that locates snakes that locally minimize this function.

4.3 Image Energy

Image energy is based on the direction of the snake with relation to the smooth spatio-temporal derivatives. First the direction of the snake as a whole is calculate as a unit vector, denoted as $\vec{v}(s)$. We then define a score function over the entire image where

$$M_{i,j} = \vec{v}(s) \cdot < f_x(i,j), f_y(i,j) > .$$

This yields a score image where each element represents the degree to which the optic flow is in alignment with $\vec{v}(s)$. Figure 4.1 depicts examples of the response in this type of matrix.



a b

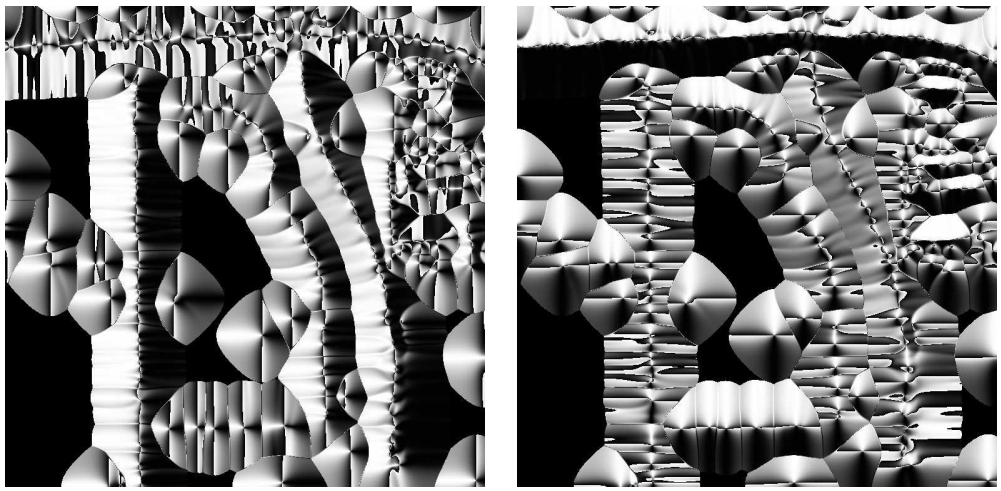
Figure 4.1: Snake Orientation Correspondences. (a) the matrix produced by a snake where $\vec{v}(s) = \langle 1, 0 \rangle$ (b) the matrix produced by a snake where $\vec{v}(s) =< 0, 1 >$. Note that horizontal roads are emphasized in (a) and vertical roads are emphasized in (b). The high responses at the bottom of the vertical road on the right are due to errors in the spatio-temporal derivatives from cropping effects in the geo-rectification.

4.3.1 Gradient Vector Flow

After the snake is seeded, an optimization process moves the snake according the the vector field produced by the energy function until the energy along the snake is minimized. This is potentially slow or incorrect because the image energy may not be non-zero at all places on the image, as depicted in figure 4.1. This problem has been addressed by Xu and Prince by using a vector propagation technique called gradient vector flow [19]. The gradient vector flow makes a vector field indicating the gradient of energy function, which serves to indicate the direction the control points should move to minimize image energy. Due to the computational intensity of the gradient flow algorithm, we use an approximated method to achieve the same effect.

1. For ∇_x, ∇_y,
2. Loop(1 to 100)
3. Let element in $\nabla_{x,y} = \frac{\nabla_i(x-1,y)+\nabla_i(x+1,y)+\nabla_i(x,y-1)+\nabla_i(x,y+1)}{5} + \nabla_i(x, y)$.

4. Upon loop termination, $\nabla_i(x,y) = \frac{\nabla_i(x,y)}{\sqrt{\nabla_x^2(x,y)+\nabla_y^2(x,y)}}$



a b

Figure 4.2: Propagated vector components. (a) propagated x-components of the dot products of the optic flow and a snake with orientation $\langle .99, .08 \rangle$. (b) the y-components.

Note that it is not necessary to extend the gradient to the edges of the image. Because the initial vectors are calculated from points in the I_t mask, and the snake itself is comprised of point from the same set, the vector propagation does not need to extend to a region far beyond that of the I_t mask.

4.4 Post-Optimization Snake Fitness

After optimization a fitness metric is applied to each snake to ensure that it is an accurate representation. Commonly, rotation in the image energy vector field will cause any snake affected by it to distort significantly. Given this, we define the following metric for fitness:

$$\sum_s |E_{bend}|,$$

which reflects to the total curvature of the snake. Using this, we define a threshold to separate fit from unfit. After empirical testing, we found that 30 is a good threshold, allowing for curved roads and for right angled turns, but removing snakes which were

significantly distorted. Upon removing snakes from the image, we also remove a small area including and around the snake from the I_t mask to prevent further snake initialization in the region.

4.5 Snake Finalization and Generation

After the termination condition [1] for snake energy minimization is met, a large area, 10 pixels, around the snake is removed from the I_t mask. Because snakes are randomly generated, we remove a large area to promote a better distribution of snakes, which yields a more complete road model. However, because a single pass of snake generation is unlikely to extract all roads, we use an iterative method of snake generation. Once no more snakes are created on the image because of the lack of seed points, then the original state of the I_t mask is restored, thereby recreating the initial conditions. The previously generated snakes remain at their same location while subsequently generated snakes are placed on the image. We have found after empirical testing that the majority of extractable roads can be found after only a few iterations of the process.

[1] This approach uses 200 updating iterations before terminating.

Chapter 5

Experiments and Results

5.1 Data Sets

Comparative studies for the algorithm are difficult because there currently does not exist a standardized aerial video test set for road identification and extraction. Furthermore, as the first algorithm to use video data, comparisons with results from static image analysis would be unfair because of the dissimilar data. Therefore, this approach has been tested using two video sequences which typify the two most common environments in which road extraction would be used and a third which tests the approach on easily acquirable data. These scenes include multiple characteristic problems and anomalies that make road extraction difficult.

5.1.1 City Scene

The city scene depicts a grid like road network with a major highway (or multi-lane road) along each axis. Multiple cars travel in both directions on these roads. Several other roads of various curvatures and orientations are present where over the course of the video only one car traverses the road. A parking lot in the upper right section of the video presents a challenge due to numerous curved road segments in a small area. This video has a significant amount of image variation caused by overhead cloud movement. Furthermore, stabilizing this aerial video required warping the original image to eliminate background motion. At times the warping could not use all the

Figure 5.1: A sample frame from the urban scene with 768x768 video resolution.

data from the original image thereby causing black borders at the edges, which affect the spatio-temporal derivatives.

5.1.2 Rural Scene

Figure 5.2: A sample frame from the rural scene with 768x768 video resolution.

The rural scene depicts a sparse road network where one car traverses a low contrast road. In the lower left section, two highways appear in the image with a diagonal orientation. To the right of the highways is a canal filled with water that becomes illuminated as the sun passes overhead due to the rotation of the aerial vehicle. Several holding towers are present in the upper right section. These towers are tall enough that georeferencing causes a significant amount of residual motion.

Furthermore, at certain locations, the car and the low-contrast road appear similar, which is depicted in figure 5.3.

Figure 5.3: An example of the low contrast between car and background in the Rural Scene, shown at 4X magnification. Note that this is the only car that traverses the major road in the video.

5.1.3 Rooftop Mall Scene

Figure 5.4: A sample frame from the rooftop mall scene with 360x360 video resolution.

This video presents a special case in that it was taken from the top of a tall building, thus giving a partial overhead view of a road scene. It depicts an intersection next to a mall, with a horizontal multi-lane road in the middle of the frame and a parking lot in the upper half. Since it was taken from a stabilized camera, there are no residual motions from rectification. Several key features of the scene are the numerous hues of car color, which affect the derivatives, and the slow speed of the cars in the parking lot. This video was included for the purposes of testing the algorithm from ground located cameras.

5.2 Results

While it is difficult to quantify the success of this approach, examination of the final results of road extraction in figures 5.6, 5.8 and 5.10 *prima facie* show the major roads and many secondary roads fitted closely by a snake.

The initial placement of snakes often yields accurate results in that the snakes are on roads and are in the general direction of the fitted road. The multiple parallel snakes depicted in figures 5.5a, 5.7a and 5.9a show the inability of the initial control point placement to form a single contour that fits to a road. Snake convergence reduces the many parallel snakes into one contour that models the roads actual shape. However, in many cases, as depicted by figures 5.5b, 5.7b and **??**b, snakes converge to non road locations. Furthermore, when snakes intersect themselves, several assumptions in our implementation of the energies break down, thereby causing incorrect forces to be applied to the control points, which exacerbates the excessive curvature of the snake. However the curvature constraint eliminates most non-road snakes. The final results are an accurate depiction of the roads as splines. Although there are some discrepancies when snakes are adjacent to the edges of the frame, the elimination of the outliers produces a consistent, uniform spline formed by overlapping snakes. The most deviation from roads occurs in the rooftop mall scene.

Examining the cases where the algorithm failed, we see that the residual motion of the towers in the rural scene causes deviations in the snake path. The road surrounding the towers accurately depicts the path of the single car, but tower motion caused a local minimum between the two towers, thereby forming a relatively straight snake that was seen as fit. We are uncertain if there is a manner to correct for this, within this analysis framework, other than observing more motion in the area to ensure a correct optic flow.

In the urban scene, the multiple lanes of traffic in a close area cause snake placement and optimization issues. During the running of the algorithm, multiple snakes were initialized to small lanes of the parking lot only to be wildly distorted by the uneven vector field. Work to correct for such an area is discussed in the further research section.

In the rooftop mall scene, the large width of the major horizontal road led to multiple parallel snakes. Furthermore, due to some camera jitter, a consistent optic

flow appeared at the roof line of one side of the mall, thereby causing a snake to identify it as a road. Since the roofline was straight, the post-optimization heuristic did not reject the snake.

Lastly, as expected, the algorithm did not extract roads where no motion was witnessed. While this is a shortcoming of the approach as a whole, we argue that this is also a problem of the data set, and that it is possible to collect data during higher traffic hours, which would maximize the number of roads extracted. Section 5.3.4 outlines an approach for merging motion based extraction with more traditional approaches which could correct this problem without the need for high-traffic data.

5.3 Further Research

While the results are encouraging, further work could be done to improve the resulting model and the output of the algorithm itself.

5.3.1 Fitness Evaluation

The post optimization step described in section 4.4 uses a simple heuristic to determine if a snake has converged to a non-road minimum. This fitness heuristic was empirically motivated by the types of snakes that were anomalies to the road model. A better function might include consistency and specificity measures computed from the eigen values of Σ. A global model of the consistency and specificity for roads in a scene could be built up as new roads are discovered, thereby yielding an evolving fitness function.

5.3.2 Formal Representation

Upon termination, roads were represented as a series of overlapping snake segments. It is unclear what the most descriptive formal model of roads should be used as a final output. As a future work, we motivate the merging of smaller snakes into a large spline that follows one road throughout the scene. A representation of intersection would complete a model for a scene. Furthermore, the spatio-temporal data could be used to identify other road features such as the number and average speed of cars,

stop lights or signs, or overpasses, all of which for static approaches are unavailable or require fine-scale image feature recognition.

5.3.3 Varied Snake Sizes

Due to the computational infeasibility of finding global minima, small snakes were used to final local minima. We define size in terms of the number of control points as well as the maximum distance between control points during the initial placement. The size of the snakes in this approach was fixed, which was hand optimized to fit the size of the majority of road segments in all scenes. However, the fixed size had the side effect of preventing the snakes from locating small road segments, such as the lanes of the parking lot in the Urban Scene data set, due to the snakes bending too much after extending past a small lane. However, it is uncertain whether allowing for varied sized snakes would locate these features without identifying other anomalous local minima. We motivate the idea of decrementing the snake size as the algorithm progress. First, large road features would be extracted using snakes with more control points, and then smaller snakes with fewer control points, or possibly restricted growth during initialization, to find small local minima from the remaining point set.

5.3.4 Motion Cues

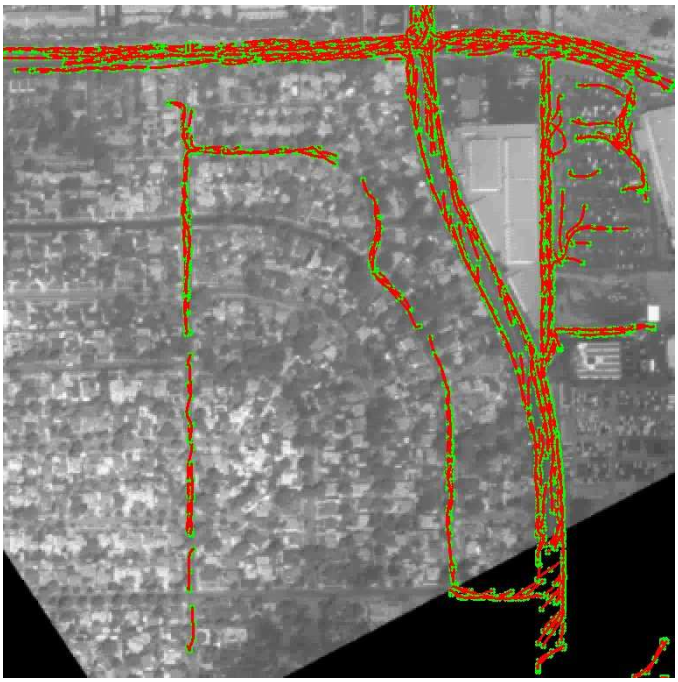
Motion cues can be used in a complimentary manner to static image analysis. As discussed in chapter 2, the different appearance of image features makes building a consistent model of road difficult. However, motion cues are consistent but sparse. To combine features of both approaches, motion cues can be used to identify salient road appearances which can therefore be used to locate roads with similar appearances that may or may not have consistent motion. This compound approach could therefore be agnostic about the actual scene specific appearance of roads and then use sparse motion cues to build its global model. This approach also has the advantage of needed only enough video to build consistent motion cues for parts of the scene to build a global model, which has the advantage over an approach that need motion cues for all roads in the scene in order.

5.4 Conclusion

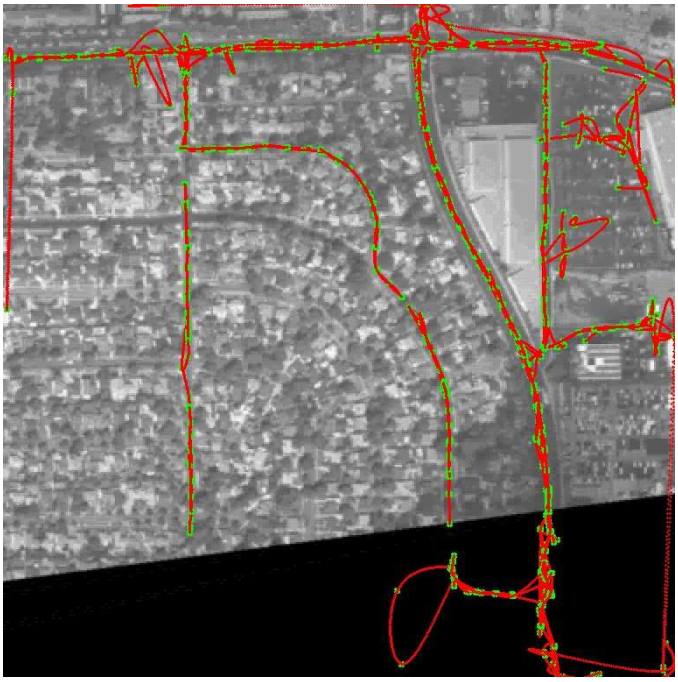
Although researchers have explored road extraction from aerial images for over twenty years, the use of aerial video and motion cues has not been available as a tool until recently. Using these, we develop a method that uses motion cues in the form of spatio-temporal derivative distributions extracted from aerial video as the basis for fitting active contours to road segments.

After discussing the necessary steps to extract the spatio-temporal derivatives, and the post-processing techniques necessary to reduce error, we presented an algorithm for identifying local segments of road that have consistent motion and matched typical road shapes. We then test the fitness of the contours that define local segments and build a global outline of the road network from contours that have above a certain measure of fitness.

The results of this approach suggest that motion cues are an excellent source of information for road extraction. Further test sets are needed to explore the additional improvements, but in the current implementation, automatic extraction is capable of identifying motion significant features with minimal error.



a



b

Figure 5.5: Roads extracted from the urban scene. (a) control points upon initial placement (b) roads after optimization of control points without a post-optimization fitness selection

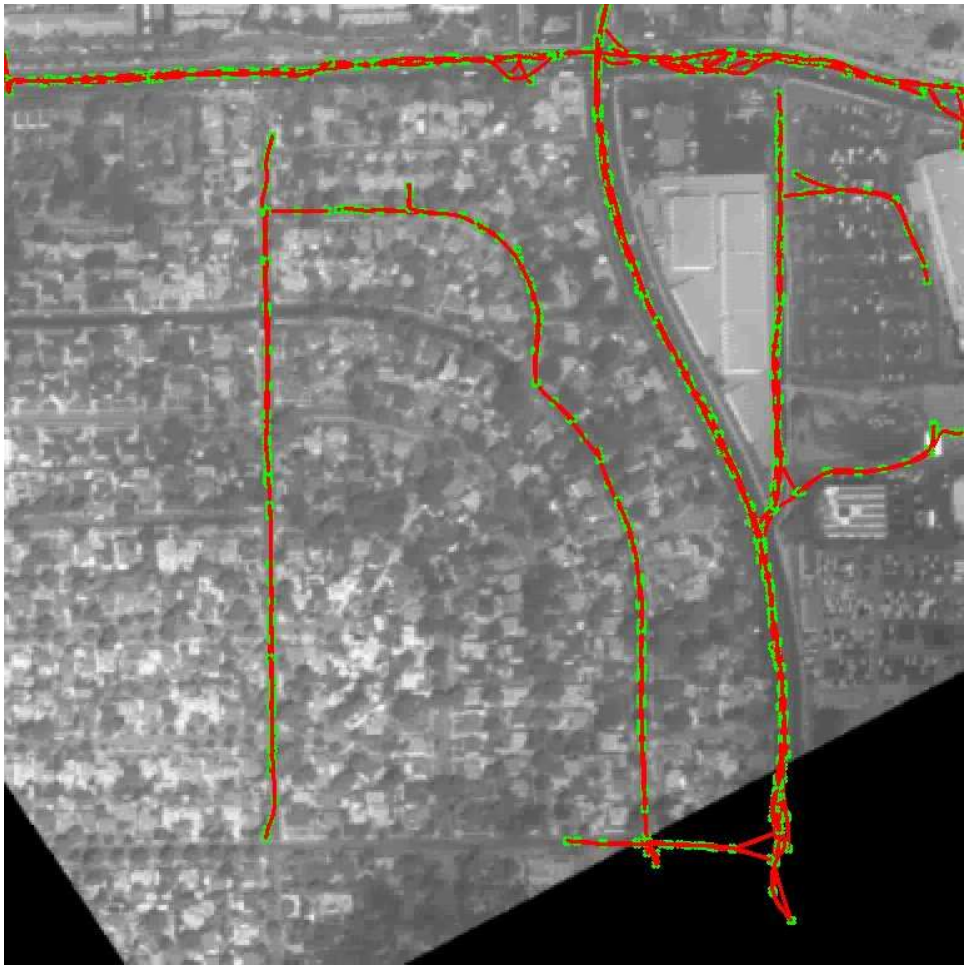


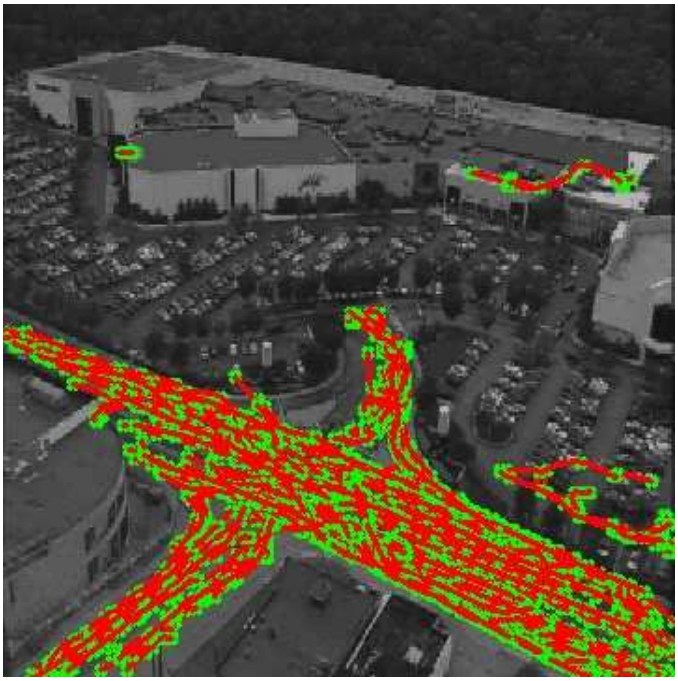
Figure 5.6: Roads in the urban scene after filtering for high bending-energy snakes.

a

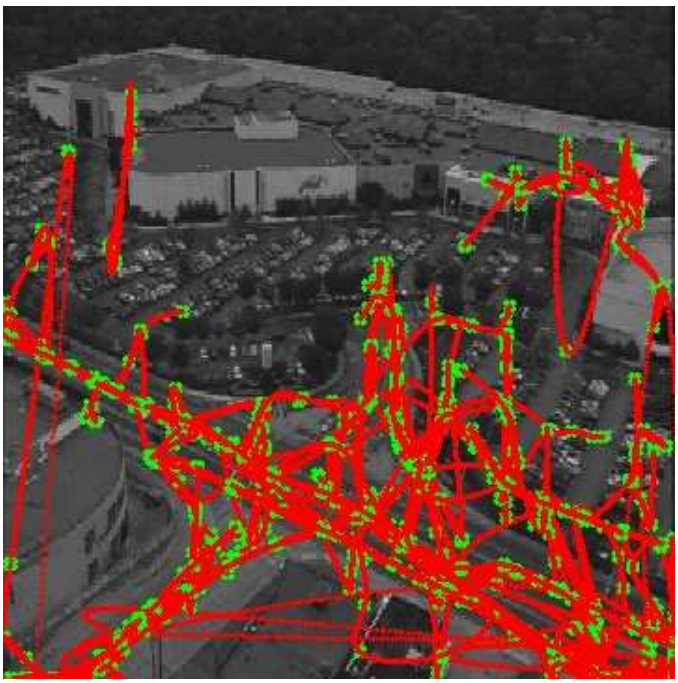
b

Figure 5.7: Roads extracted from the rural scene. (a) control points upon initial placement (b) roads after optimization of control points without a post-optimization fitness selection

Figure 5.8: Roads in the rural scene after filtering for high bending-energy snakes



a



b

Figure 5.9: Roads extracted from the rooftop mall scene. (a) control points upon initial placement (b) roads after optimization of control points without a post-optimization fitness selection

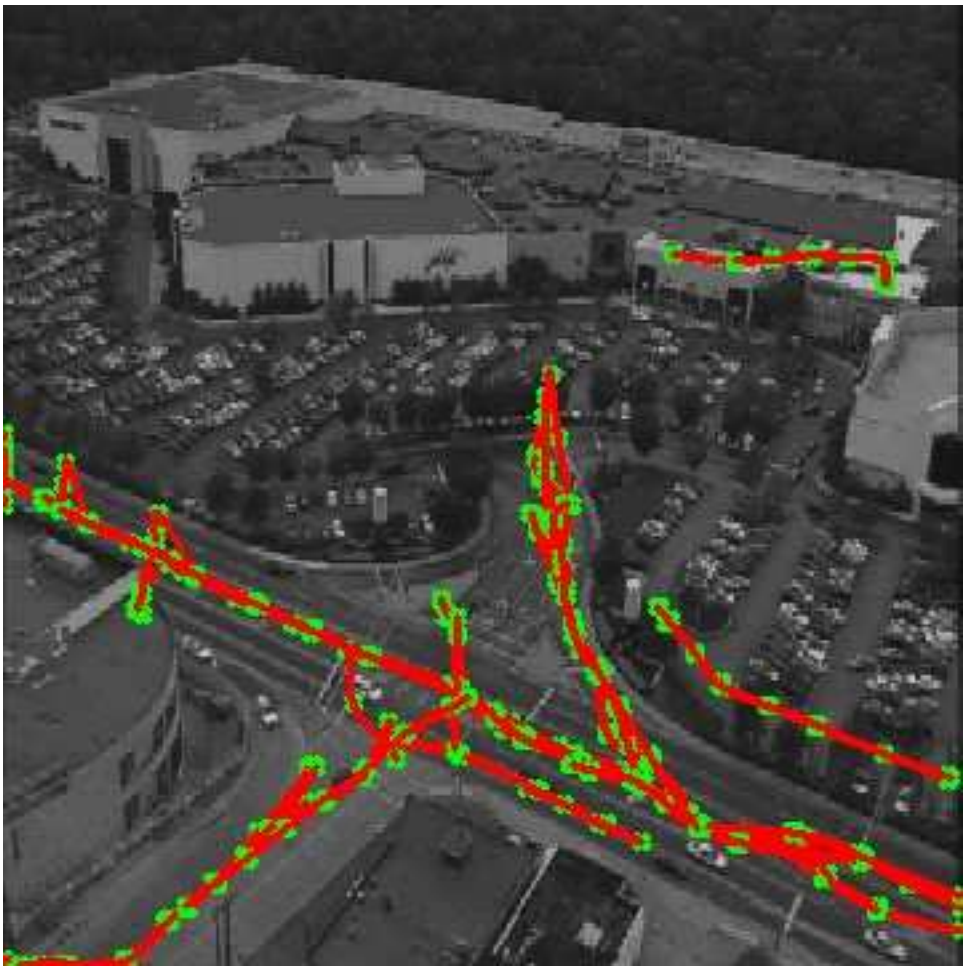


Figure 5.10: Roads in the rooftop mall scene after filtering for high bending-energy snakes

Appendix A

Full Pseudocode for the Algorithm

This section outlines the major methods in the algorithmic approach described in this paper. This pseudo code assumes that the programmer has access to working subroutines capable of performing operations such as Sobel filtering or single value decomposition. Such functions are common in many image processing libraries, such as the Intel OpenCV library which is used to generate the results in this thesis, and therefore discussion of how to perform such functions is not seen as necessary.

A.1 Calculating The Spatio-Temporal Derivatives

For all frames

Blur using a Gaussian blur kernel.

For all pixels (x, y) on frame i

$$\text{Let } I_x = \text{the Sobel filter response of } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ at } (x, y)$$

$$\text{Let } I_y = \text{the Sobel filter response of } \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ at } (x, y)$$

$$\text{Let } I_t = \begin{cases} I_i(x, y) - I_{i-1}(x, y) & i > 0 \\ 0 & i = 0 \end{cases}$$

Store the free parameters of the coviance matrix, where the parameters have been initialized to zero on the first iteration.

$I_xI_x+ = I_x \cdot I_x$

$I_xI_y+ = I_x \cdot I_y$

$I_xI_t+ = I_x \cdot I_t$

$I_yI_y+ = I_y \cdot I_y$

$I_yI_t+ = I_y \cdot I_t$

$I_tI_t+ = I_t \cdot I_t$

A.2 Computing The Optic Flow

For the covariance matrix Σ at pixel (x, y), compute the single value decomponsition (SVD) of Σ, with the resulting matrix denoted as S.

The optic flow is the third eigenvector of the solution e_3, with u and v being normalized by the third component of the eigen vector.

Therefore, $u = \frac{e_3(1)}{e_3(3)}, v = \frac{e_3(2)}{e_3(3)}$.

A.3 Covariance Matrix Smoothing

For the covariance matrix $\Sigma = \begin{bmatrix} I_xI_x & I_xI_y & I_xI_t \\ I_xI_y & I_yI_y & I_yI_t \\ I_xI_t & I_yI_t & I_tI_t \end{bmatrix}$ at pixel (x, y)

Initialize to 0 a new matrix $\hat{\Sigma}$ that will be the result.

Let $< u, v >$ be the normalized optic flow vector (x, y)

Let $T = \begin{bmatrix} 30 \cdot u & 30 \cdot v \\ -3 \cdot v & 3 \cdot u \end{bmatrix}$

Let $M = T^\top T$

For pixel (i, j), let the weight of $\Sigma_{i,j}$ be $w = e^{-<i,j>^\top M^{-1} <i,j>}$

For $i = -5$ to 5, $j = -5$ to 5,

$\hat{\Sigma}$+=$w \cdot \Sigma_($$x + i, y + j)$, where + denotes the element-wise addition of the matrices (i.e. $M[i, j]$+=$N[i, j]$)

A.4 Generating The I_t Mask

Initialize matrix $mask$ to the size of a frame.

For all frames

For each pixel (x, y)

If $|I_t| > threshold$

$mask[x, y] = max(mask[x, y], |I_t|)$

for $i = x - 2$ to $x + 3$; for $j = y - 2$ to $y + 3$

$mask[i, j] = max(mask[i, j], |I_t|)$

A.5 Snake Initialization

Choose a random point (x, y) in the I_t point set.

Initialize a set of control points with a fixed size

Set the first control point to (x, y)

Choose the remaining control points as follows:

For control point i

Let $< u, v >$ be the normalized optic flow at control point $i - 1$

Let θ be the angle at control point $i - 1$

Let the coordinates for control point i (x_i, y_i) be intialized to (x_{i-1}, y_{i-1}).

For scale factor s, 1 to 20

$i = x + s \cdot u, j = y + s \cdot v$ and if (i, j) is in the I_t point set and if the dot product of $< u, v >$ and the normalized optic flow vector at (i, j) is greater than .75

$(x_i, y_i) = (i, j)$

else

for angle offset δ from $\pm\frac{\pi}{4 \cdot s}$ to $\pm\frac{\pi}{4}$

$i = x + (s \cdot \cos(\theta + \delta))$

$j = y + (s \cdot \sin(\theta + \delta))$

if (i, j) is in the I_t point set and if the dot product of $< u, v >$ and the normalized optic flow vector at (i, j) is greater than .75

$(x_i, y_i) = (i, j)$

break

If the number of control points with the same location is $\geq \frac{1}{2}$

If the percentage is $\geq \frac{3}{4}$

Remove points in the I_t mask with a distance ≤ 5 from the snake.

else

Remove points in the I_t mask with a distance ≤ 1 from the snake.

A.6 Snake Optimization

For a snake constructed of control points denoted $cp(i)$

Check whether the snake is valid by counting the number of control points that are at the same position.

If the snake is invalid due to a high number of identical points

Remove the section of the I_t point set where the invalid snake was placed.

Compute the direction of the snake as a unit vector $< u, v >$

Compute the matrix M where $M[x, y]$ equals the dot product of $< u, v >$ and the optic flow at (x, y).

Compute the gradient matrices X and Y by applying Sobel filters (the same descripted in computing I_x and I_y) to M where gradient

$G[x, y] = \begin{cases} Sobelfilterresponseat(x, y) & (x, y)\text{is in the } I_t \text{ mask point set} \\ 0 & (x, y)\text{is not} \end{cases}$

Looping 1 to 100

For each gradient, $G[x, y] = \frac{G[x-1,y]+G[x+1,y]+G[x,y-1]+G[x,y+1]}{4} + G[x, y]$

Note that it might be necessary to rescale all the values in G values if overflow occurs

On last loop iteration normalize the gradient vector, $G[x, y] = \frac{G[x,y]}{\sqrt{X[x,y]^2+Y[x,y]^2}}$

Optimize the control points

For the desired number of loops

Let the total curvature be a vector initilized to $< 0, 0 >$.

For each control point i

Let (x, y) denote the position of i on the image

Compute the image, expansion and bending energy as vectors

Add the absolute values of of the bending energy vector components to the corresponding components of the total curvature vector

Image energy $=< X[x, y], Y[x, y] >$

Bending Energy $=< \frac{cp_{i-1}(x)+cp_{i+1}(x)}{2} - x, \frac{cp_{i-1}(y)+cp_{i+1}(y)}{2} - y >$ where $cp_i(x)$ denotes the x coordinate of the i^{th} control point.

Expansion Energy $= \frac{<x-cp_{i-1}(x), y-cp_{i-1}(y)>}{.2\cdot\sqrt{(x-cp_{i-1}(x))^2,(y-cp_{i-1}(y))^2}+1} + \frac{<cp_{i+1}(x)-x, cp_{i+1}(y)-y>}{.2\cdot\sqrt{(cp+i-1(x)-x)^2,(cp_{i+1}(y)-y)^2}+1}$

Choose weight parameters, α, β and γ

For point i, $(x', y') = (x+\alpha E_{img}(x)+\beta E_{bend}(x)+\gamma E_{exp}(x), y+\alpha E_{img}(y)+\beta E_{bend}(y) + \gamma E_{exp}(y))$

If the length of the total curvature vector is greater than 30

Remove points in the I_t mask with a distance ≤ 2 from the snake.

Upon snake finalization, remove points in the I_t mask with a distance ≤ 10 from the snake.

A.7 Execution Outline

For all frames of the video

Calculate the spatio-temporal derivatives

Update the I_t Mask

Compute the optic flow

Perform covariance matrix smoothing

Recompute the optic flow from the smoothed covariance matrices

Store a copy of the intial state of the I_t mask

Loop for the number of snake generation iterations

Loop for the number of snakes

Initialize Snake

Optimize snake

Reset the current I_t mask to its initial state

Output video frame with snakes

References

[1] M. Bicego, S. Dalfini, G. Vernazz, and V. Murino. Automatic road extraction from aerial images by probabilistic contour tracking. In *Proceedings of IEEE International Conferences on Image Processing (ICIP03)*, volume III, pages 585–588, 2003.

[2] X.-T. Dai, L. Lu, and G. Hager. Real-time video mosaicing with adaptive parameterized warping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001. Demo program.

[3] H. Farid and E. P. Simoncelli. Optimally rotation-equivalent directional derivative kernels. *Computer Analysis of Images and Patterns (CAIP)*, 1997.

[4] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996.

[5] S. Hinz and A. Baumgartner. Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58:83–98, 2003.

[6] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artifical Intelligence*, 17:185–203, 1981.

[7] Reinhold Huber and Konrad Lang. Road extraction from high-resolution airborne SAR using operator fusion. In *Proceedings of International Geoscience and Remote Sensing Symposium*, July 2001.

[8] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, 1991.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. In *First International Conference on Computer Vision*, pages 259–268, 1987.

[10] G. Kühne, J. Weickert, O. Schuster, and S. Richter. A tensor-driven active contour model for moving object segmentation. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 73–76, 2001.

[11] Jun Kumagi, Huijung Zhao, Masafumi Nakagawa, and Ryosuke Shibasaki. Road extraction from high-resolution commercial satellite data. In *22nd Asian Conference on Remote Sensing*, 2001.

[12] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner. Automatic extraction of roads from aerial images based on scale-space and snakes. *Machine Vision and Applications*, 12(1):23–31, 2000.

[13] Ivan Laptev. Road extraction based on snakes and sophisticated line extraction. Master's thesis, Royal Institute of Technology, Stockholm, Sweden, 1997.

[14] Cheng-Yi Lin and Chi-Farn Chen. Automated extraction of control points for high spatial resolution satellite images. In *22nd Asian Conference on Remote Sensing*, 2001.

[15] Seung-Ran Park and Tsejung Kim. Semi-automatic road extraction algorithm from IKONOS images using template matching. In *22nd Asian Conference on Remote Sensing*, 2001.

[16] R. Pless, T. Brodsky, and Y. Aloimonos. Detecting independent motion: The statistics of temporal continuity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8):768–773, 2002.

[17] F.M. Porikli. Road extraction by point-wise gaussian models. *SPIE AeroSense Technologies and Systems for Defense and Security*, 5093:758–764, 2003.

[18] K. Price. Urban street grid description and verification. In *IEEE Workshop on Applications of Computer Vision*, pages 148–154, 2000.

[19] C. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

Vita

David A. Jurgens

Date of Birth	October 26, 1981
Place of Birth	Berwin, Illinois
Degrees	B.A. College Honors, Philosophy, May 2004
Professional Societies	Association for Computing Machines
Publications	Robert Pless and David Jurgens, "Road extraction from motion cues in aerial video." Submitted to ACMGIS 2004, May 2004.

December 2004